Pearson Software Consulting, LLC

Array Formulas

Array Formulas are formulas that work with arrays, instead of individual numbers, as arguments to the functions that make up the formula.

Introduction To Array Formulas

When an array formula is displayed, it is surrounded by braces {}. You do not enter the braces. Instead, when you enter an array formula, you press `Ctrl+Shift+Enter`, rather than just `Enter`. Excel will automatically add the braces. You must press `Ctrl+Shift+Enter` when you first enter an array formula, and also each time you edit the formula later. If you enter or array formula without pressing `Ctrl+Shift+Enter`, it will return an incorrect result or a `#VALUE!` error.

For example, consider the formula

```
=IF(A1=B1,1,0)
```

This will return either 1, if `A1=B1`, or 0 if they are not equal. But suppose you want to get a count of
the number of cells in a range, `A1:A10`, which are equal to the corresponding cells in `B1:B10`. By using an array formula, you can do this with a single formula by passing arrays to the `=IF` function, and summing up the results:

```
=SUM(IF(A1:A10=B1:B10,1,0))
```

By entering this as an array formula, you are instructing the `=IF` function to "loop" through the range `A1:A10`, compare each element to the corresponding element in `B1:B10`, and return an array of 1's and 0's, each element of which indicates the result of each comparison. The `=SUM` function adds up this array, and returns a single number indicating the number of cells in `A1:A10` which are equal to their counterparts in `B1:B10`.

When you use more than one range in an array formula, all of the ranges *must* contain the same number of elements. Otherwise, an error is returned.

Array formulas are ideal for

counting cells in a range with multiple criteria. For example, suppose in `A2:A10` we have a product-name, in `B2:B10` we have a salesman-name, and in `C2:C10` we have number of units sold:

	A	B	C	
1	*Product*	*Salesman*	*Units Sold*	
2	Fax	Brown	1	
3	Phone	Smith	10	
4	Fax	Jones	20	
5	Fax	Smith	30	
6	Phone	Jones	40	
7	PC	Smith	50	
8	Fax	Brown	60	
9	Phone	Davis	70	
10	PC	Jones	80	

To compute the number of *Phones* sold by *Smith*, we would use the following array formula:

```
=SUM
((A2:A10="Phone")*
(B2:B10="Smith")
*C2:C10)
```

This formula works by looping, looking at the elements in the three ranges `A2:A10`, `B2:B10`, and `C2:C10`. These three ranges do not have to be in adjacent columns, or even in the same rows, but they *must* contain the same *number* of rows.

The ranges are looped though "simultaneously." Element `Ai` is evaluated in the same iteration as elements `Bi` and `Ci`.

If element `Ai` is *Phone*, a `1` (`True`) is returned. Otherwise, `0` (`False`) is returned. If element `Bi` is *Smith*, a `1` is returned, otherwise `0` is returned. Then `Ci` is returned. These three values are multiplied together. The product will be either `0` (if either or both of the `Ai` or `Bi` comparisons were false) or `Ci`. Summing these results together gives of the sum of entries `Ci` where `Ai` is "Phone" and `Bi` is "Smith".

Logical Operations With Array Formulas

In addition to the AND operation described above, you can also use array formulas to perform an OR, and XOR, or even a NAND (Not AND) operation.

Addition simulates the OR operation. If either one or both conditions are true, addition will return a non-zero result. Using the example above, we can count the number of sales, in which either a Fax was sold, or Jones was the salesman (or both).

```
=SUM(IF((A2:A10="Fax")+(B2:B10="Jones"),1,0))
```

```
=SUM(IF(MOD((A2:A10="Fax")+
(B2:B10="Jones"),2),1,0))
```

The MOD operator in the above formula returns 0 when either both conditions (Fax, Jones) are True or when both are False. It returns 1 only when either exactly one of the conditions is True.

NAND is an Negative And operation, which returns true when neither or one of the elements is True. It returns False when BOTH elements are True.

```
=SUM(IF((A2:A10="Fax")+(B2:B10="Jones")<>2,1,0))
```

This counts all sales, except those in which Jones sold a Fax.

You can combine these formulas to create rather complex logical tests. Just be careful to make sure that you parentheses are in the right places.

Formulas That Return Arrays

The other "flavor" of array formulas are functions that return arrays *as their result.* To work with these types of functions, you must enter the same formula into an array of cells. As is often the case, an example best explains this.

Suppose we have a matrix in `A1:C3`. We can use the **`=MINVERSE`** function to return the inverse of this matrix. Since the inverse of a matrix is itself a matrix, we've got to enter the **`=MINVERSE`** function as an array formula.

Select a range of cells, `A5:C7`, in our example, with the same number of rows and columns as the original matrix, enter **`=MINVERSE(A1:C3)`**, and press `Ctrl+Shift+Enter`. This enters the `=MINVERSE`

	A	B	C
1	2	0	1
2	0	2	1
3	1	2	0
4			
5	0.33	-0.33	0.33
6	-0.17	0.17	0.33

function as an array formula into all of the selected cells, and the inverse matrix will be returned into this array. You'll notice that when you enter a formula into an array of cells, you cannot alter a single cell in the array. You must edit or delete the entire array.